

ARMY RESEARCH LABORATORY



Turbine Engine Diagnostics (TED): A Practical Application of a Diagnostic Expert System

by Holly Ingham, Richard Helfman, Timothy Hanratty,
John Dumer, and Edmund H. Baur

ARL-SR-60

January 1998

19980212 083

DTIC QUALITY INSPECTED 4

Approved for public release; distribution is unlimited.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-SR-60

January 1998

Turbine Engine Diagnostics (TED): A Practical Application of a Diagnostic Expert System

Holly Ingham, Richard Helfman, Timothy Hanratty, John Dumer,
Edmund H. Baur
Information Science and Technology Directorate, ARL

Abstract

Turbine engine diagnostics (TED) is a diagnostic expert system that aids the M1 Abrams' mechanic in finding and fixing problems in the AGT 1500 turbine engine. TED was designed to provide the apprentice mechanic the ability to diagnose and repair the turbine engine like an expert mechanic. This report discusses the reasoning method used in TED, called the procedural reasoning system (PRS), as well as various design considerations throughout the life of the project. The expert system was designed and built by the U.S. Army Research Laboratory (ARL) and the U.S. Army Ordnance Center and School (USAOC&S). TED has been fielded to both the Active Army and the National Guard.

Table of Contents

	<u>Page</u>
List of Figures	v
1. Introduction	1
2. Reasoning in TED	2
3. Developmental Issues	5
3.1 Establish and Maintain Communication	6
3.2 Rapid Prototyping	7
3.3 Spiral Model	7
4. TED Software Overview	8
4.1 Design Goals	8
4.2 Software Selection	9
4.3 Soldier Interface	11
5. TED's Future Direction	11
6. References	13
Distribution List	15
Report Documentation Page	17

INTENTIONALLY LEFT BLANK.

List of Figures

<u>Figure</u>	<u>Page</u>
1. PRS Architecture	3
2. VX Procedure in TED	4
3. Spiral Model	8
4. Sample TED Menu	12

INTENTIONALLY LEFT BLANK.

1. Introduction

Expert systems development has become the most successful application area of artificial intelligence (AI). Numerous systems have been developed and applied in areas ranging from law to forestry to robotics [1-3]. Many new tools and techniques have been developed that add to the increased interest and success of these expert systems. As an example of these tools and techniques being fostered in the area of expert systems, this report will focus on turbine engine diagnostics (TED), a diagnostic expert system to aid the M1 Abrams' mechanic in finding and fixing problems in the AGT-1500 turbine engine. TED was designed and built by the U.S. Army Research Laboratory (ARL) and the U.S. Army Ordnance Center and School (USAOC&S). Limited fielding was begun in July 1994 to selected National Guard units, and it is currently being fielded to active units of the U.S. Army. TED was designed to provide the apprentice mechanic the ability to diagnose and repair the turbine engine like an expert mechanic.

TED focused on the maintenance of the AGT-1500 engine due to the overwhelming repair costs of this engine to the Army. The M1 Abrams tank is the Army's main weapon system, with over 7,500 tanks fielded to active and reserve units. The repair costs were noted in a study that concluded, "the maintenance costs of the AGT-1500 engine represents the largest portion of the Army AGT-1500 operation and support (O&S) costs" [4]. Another study determined that in 1 yr, out of a group of 360 engines evacuated to depot, 39% of them were reported as no evidence of failure (NEOF). The NEOF condition means that an engine was pulled from the tank and sent back to the depot for repair, but the depot determined there was nothing wrong with the engine. The unnecessary cost related to NEOF conditions was estimated at \$18.2 million annually [5]. The development and fielding of the TED program represents the Army's first successful fielded maintenance system in the area of AI. There are several reasons associated with the success of the TED program: appropriate domain with proper scope, a close relationship with the expert, extensive user involvement, plus others that will be discussed later in this report.

2. Reasoning in TED

The main diagnostic software in TED is a Windows-based expert system shell called Visual Expert (formerly Adept) from SoftSell Technology Inc. Visual Expert is based on a reasoning paradigm called procedural reasoning system (PRS) [6, 7]. PRS is a visual method of encoding reasoning strategies used by expert problem solvers. The knowledge is represented graphically with semantics suited to the procedural, goal-oriented style of problem solving. PRS is best suited for problems that are both procedural and goal oriented. A procedural approach uses an ordered step-by-step prescription to obtain a desired result, possibly including alternate paths in case of failure. Such an approach is also goal oriented if some steps are goals to be achieved rather than specific actions to be performed [8].

PRS is endowed with the attitudes of belief, desire, and intention (see Figure 1). The generalized system is composed of a system database, a set of procedures or plans, an interpreter or inference engine, and a process stack. The database contains the current beliefs of the system. These beliefs could be static properties of the domain or beliefs derived by the system itself as it executes its plans. The plans are descriptions of how to accomplish given goals or to react to certain situations and are represented by declarative procedure specifications. The body of these procedures is represented as a graphical network with sequences of subgoals to be achieved as well as primitive actions to be accomplished. The interpreter runs the entire system, executing active goals and deciding what course of action to take based on the beliefs the system has at a point in time.

Visual Expert was chosen as the primary tool for development due to TED's need for rapid prototyping, a failed first attempt using a rule-based reasoning system, and the fact that Army technical manuals (TMs) closely follow the paradigm of the PRS. Visual Expert uses the concept of visual application creation where development takes place through the manipulation of graphical objects on the screen. This provides an environment that is well suited for rapid prototyping. A program built with Visual Expert is composed of procedures within which there are nodes that are connected via true, false, or unknown arcs. Only one procedure is viewable in its entirety at one

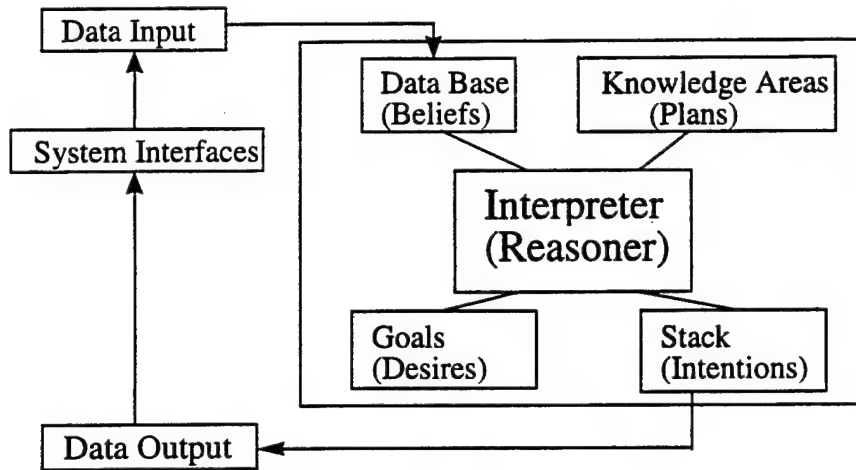


Figure 1. PRS Architecture.

time, however, all procedures in the program are viewable by name. This keeps the program environment uncluttered and organized for the programmer. The visual environment was so effective that the subject matter experts (SMEs) assigned to the TED program quickly learned to read the code and some began writing their own code or modifying code written by the knowledge engineers. Visual Expert also provides a debugging environment that further aids the rapid prototype development by shortening the find, fix, and verify time for software problems. Design flaws and/or faulty logic is also much easier to find using a visual-based development tool like Visual Expert.

Visual Expert consists of a system database, a display and procedure builder, scripting for application development, an interpreter or inference engine, and a debugger. The display builder is the mechanism by which information is presented to or received from the user. It contains all of the commonly used Microsoft Windows object building tools, such as buttons, text boxes, etc. The procedure builder provides the developer the ability to graphically represent a process for solving a problem or performing a task and is the backbone of Visual Expert applications (see Figure 2).

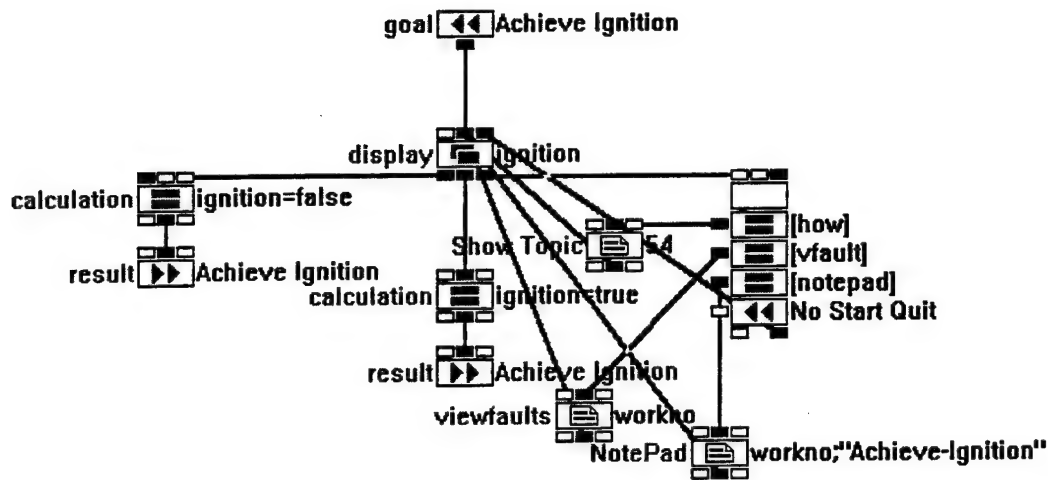


Figure 2. VX Procedure in TED.

“Procedures consist of a series of nodes, each containing a set of incoming and outgoing arc handles. Nodes are connected by arcs showing the sequence of the steps. Information is fired from one node to another through these arcs based on the state of the world. The number and arrangement of the arc handles are determined by the node’s type” [9].

There are five different node types used to represent steps in a procedure: start, work, case, end, and custom. Work nodes represent the logical flow in a true/false/unknown relationship. Nodes also have several styles: calculation, goal, display, and result. A calculation node performs a mathematical operation, calls a function, or compares variables. A goal node links to another procedure that can solve a problem and return an answer. A result node triggers another procedure, depending on the result of an operation. The display node provides the capability of developing the user interface for a program using the full suite of Microsoft Windows controls. Default displays are provided along with tools to customize displays with text, colors, graphics, shapes, buttons, list boxes, etc. Scripting provides the capability of developing nodes that behave in a way that is

particular to a specific application. A script is a collection of statements that define the behavior of a custom node. They can be as simple as a one-line statement that assigns a value to a variable to as complex as containing instructions that perform operations under different conditions.

3. Developmental Issues

The development of any large-scale computer system requires extensive amounts of time and resources. Expert systems are no exception. Careful consideration must be given to a myriad of issues. The following section outlines the critical issues that were part of the TED development process.

The principal reasons for developing an expert system are to disseminate rare or costly expertise and to more effectively and efficiently use the human expert [10]. The selection of an appropriate domain with proper scope is critical to its success. The domain selected should be one that encompasses a problem that is "worthy" of the effort. The specificity within the domain defines the scope of the project. For the TED program, Abrams tank maintenance was quickly identified as the proper domain with special focus on the engine and transmission.

By 1991, several factors were contributing to the selection of tank maintenance as an appropriate domain. In addition to rising maintenance costs, the Army had developed a new funding directive called stock funding of depot level repairables (SFDLR). Essentially, this directive puts the burden of maintenance costs onto the company commander rather than deferring it up the chain, in the hope that overall maintenance costs will be reduced. Finally, the Army had also developed a new maintenance doctrine in order to maintain a high operational readiness status. Under the new doctrine, when an engine fails, it is pulled from the tank and sent to Direct Support (DS). The tank hull remains at the unit, a new engine is sent forward, and the tank is quickly returned to full operational status.

The TED software engineers quickly established some important guidelines that remain in effect today.

3.1 Establish and Maintain Communication. Software engineers and SMEs do not generally speak the same language. Software engineers talk of frames and objects. The SMEs for the TED program are M1 tank mechanics. M1 tank mechanics talk of inlet guide vane (IGV) angles and of rotational variable differential transformers (RVDTs). Each needs to learn some of the other's language, but the main effort is on the software engineer to learn the language of the mechanic.

The best way to learn what the user does is to observe the user in his environment. The TED team attended and videotaped classes for M1 mechanics. This produced three important benefits. First, it quickly immersed the software engineers into the language of the mechanic. The IGV is located in front of the engine, and the angle determines how much air gets through to the turbine blades. Second, it gave an accurate picture of how a mechanic performs his job and how software might improve that job. The TED team noticed during that first session that the original scope of work was too narrow. There was a whole suite of software that could help the mechanic better perform his job. Third, it established a bond between the software engineer and the soldier. Soldiers could sense that the team was serious and that soldier's needs would be given serious attention. They were thus eager to cooperate.

When the aim is to produce software that not only works as planned but also gets used by the mechanic, then user participation in the development process is critical. The TED team heard many stories from soldiers about equipment that never gets used and equipment that is difficult to use for which a small change would have made the item soldier friendly. The TED SMEs were assigned full time to the project.

New technology is often met with resistance when it is thrown at an unaware and/or ill-prepared user. Rarely can a user, at the start of a project, envision how technology can improve his job. A system based on initial user expectations will at best be shallow and may even be useless. The software engineer and the SME are each constantly learning about the other. The software engineer

is continually learning about the needs and duties of the mechanic, and the mechanic is learning about the potential impact of new software on his future.

3.2 Rapid Prototyping. A prototype is essential for two-way communication. It allows the user to see and touch what the software engineer envisions for the user. It gives the user the earliest opportunity to comment on his system, and it gives him some clue as to the potential of the project. The user does not always know what technology is available, and the hands-on experience of the prototype is often the best way to educate the user. A prototype serves as a common reference point. Without a prototype, it is difficult to obtain useful feedback. It also shows how well the software engineer understands the user's needs.

3.3 Spiral Model. Boehm's spiral model [11] shown in Figure 3 incorporates an incremental development schema. Successive prototypes are produced that expand upon user requirements. In addition, the software engineer is able to break down complex tasks into smaller components. As each component is developed, it is evaluated against user requirements. The user requirements are reevaluated as each successive module is developed. Consequently, the user is an integral part of the development team. His input is essential. There are two reasons behind selecting the spiral method for the TED program: rapid changes in PC hardware and software and the need to keep the user in the loop. In 1991, it was obvious that hardware and software for the PC would continue to improve and become more affordable. Computer memory continues to expand and deflate in price. Hard drives continue to get bigger and cheaper. Screen resolution expands and video cards improve. The price of a Pentium system today rivals the price of a 386 system in 1991.

Software follows the same pattern outlined for hardware. Every year, software improves, new products are announced, and existing products offer upgrades at an astounding pace and price. Goals that were impossible or difficult in the past may now be relatively easy tasks. The TED team continues to meet formally once a month to decide on the direction and scope of the project. Unsatisfied goals are reevaluated, and some may be dropped from the list, while new goals may be added.

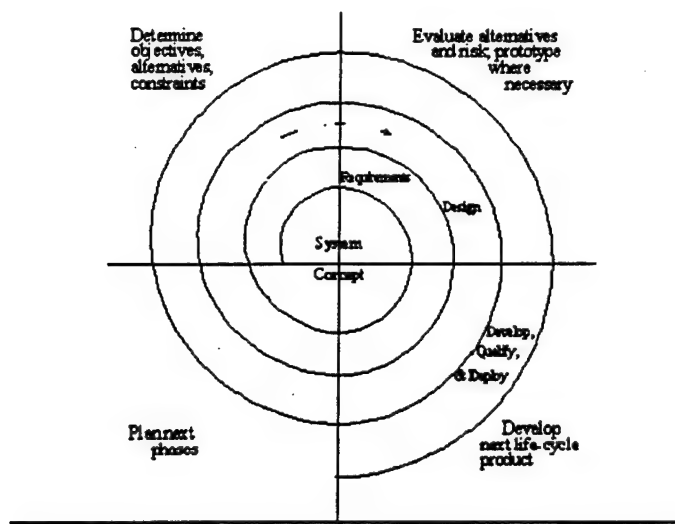


Figure 3. Spiral Model.

4. TED Software Overview

4.1 Design Goals. At about 6 mo into the project, the SMEs had established several design goals. These goals were based primarily on each SME's extensive experience as an M1 mechanic and as an M1 instructor for engine maintenance classes. The SMEs had much previous experience with soldier mechanics—their likes and their dislikes. The following lists the main design goals for the TED software. The software should be:

- accurate,
- easy to use,
- flexible,

- task oriented, and it should
- support multiple levels of expertise.

First, the software should be accurate. It need not be perfect, but it should be significantly better at diagnosing faults than the system it is replacing. Otherwise, it will lose soldier respect and it will not be used. Second, it must be easy to use, for otherwise, it will sit on the shelf. Mechanics have favorite stories of diagnostic equipment that does nothing but take up lots of storage space. Third, it must be flexible enough to support a variety of diagnostic styles. For example, some mechanics are thorough and methodical, and a structured step-by-step approach is best for them. A few have a sixth sense and know what is wrong with an engine. They have only limited need for the information in TED and will only use it as an occasional reference. Other soldiers have a mixture of styles. They may know a lot about some parts of the engine but need guidance on other areas.

The fourth goal is that TED be task oriented and structured in a way that is natural for the soldier. The current TMs have a structure that is difficult to use and to follow. For example, consider a typical task to determine whether excessive metal chips are present. To perform this check, the user must first find the right TM. It's in TM-34. Once in the right TM, the job is to find the right page. Symptom 2, Metal Chips, begins on page 3-20. The tasks for Symptom 2, Metal Chips Found, refer to tasks in TM 20-1 and in TM 34-1. However, little information is given as to which page in TM 20-1 or TM 34-1 to turn to. Experts can navigate the TMs, but others find the structure confusing.

The last goal recognizes that mechanics come with different skill levels. Experts need little or no help from TED. Beginners need extensive step-by-step instructions. A system aimed at just one level of expertise would bore the expert or baffle the beginner.

4.2 Software Selection. The Army had already chosen the hardware for TED, the Contact Test Set III (a ruggedized 486 PC), which is capable of running Unix, DOS, or Windows. It was clear from the beginning that the project would involve a variety of tasks, each needing a specialized

software package. It was also clear that no package could run in isolation. Programs would need to exchange information with others. Windows was chosen as the operating system because of its capabilities and its perceived growth potential.

For any software choice, the key is to choose a package that first meets the user's needs, and then, if possible, the programmer's. One choice the programmer must often make is whether to choose commercial off-the-shelf (COTS) packages or whether it is better to write the code himself.

Today COTS packages offer many advantages over code produced in-house. They also have some disadvantages. The benefits include:

- Cost is reduced by spreading among many.
- External support is available from the developer.
- Code is already written, saving time.
- Technology proliferation offers many selections.

The disadvantages may include:

- The program may not fit the problem.
- Program success is tied to the survivability of the COTS developer.
- Initial code may work but upgrades may not.
- Run-time fees may be high.

The TED team prefers to use COTS software when available and suitable. Whenever such software is not available or suitable, the choice is either to wait until a new product is released or a product upgrade provides the needed functionality, or write the code in-house. For example, the current hypertext package was not chosen until the fall of 1993, and the database was not selected until the fall of 1994. These code decisions are subject to change at each monthly meeting. As the team gathers experience with a package or code, the decision might be to continue as before, to switch from in-house to COTS (or vice versa), or to switch COTS vendors.

4.3 Soldier Interface. Users communicate with TED primarily through the mouse, and sometimes through keyboard input. At the top level, TED is menu driven (see Figure 4). At this level, the soldier can choose which module to run. Inside a module, TED can be either soldier driven or data driven. Soldier driven means that TED is in browse mode. This is the equivalent of opening the TM to any section and reading the pages. Browse mode is useful for experts who need little supervision and only occasional help from the TMs.

In data-driven mode, TED first reads its knowledge base (a database of faults previously identified) to determine engine history and then leads the mechanic through a series of tasks to perform and/or questions to answer. All pertinent information is linked so the user is automatically lead through different sections of the TMs, if necessary. The user can hop out of this mode at any time and jump into browse mode.

5. TED's Future Direction

As indicated earlier, TED is currently fielded in both the Active Army and the National Guard. It is the Army's first successful large-scale application of Expert System technology. Future efforts are concentrated on embedding TED into the tank itself and expanding the scope of diagnostics to other areas of the M1 tank.

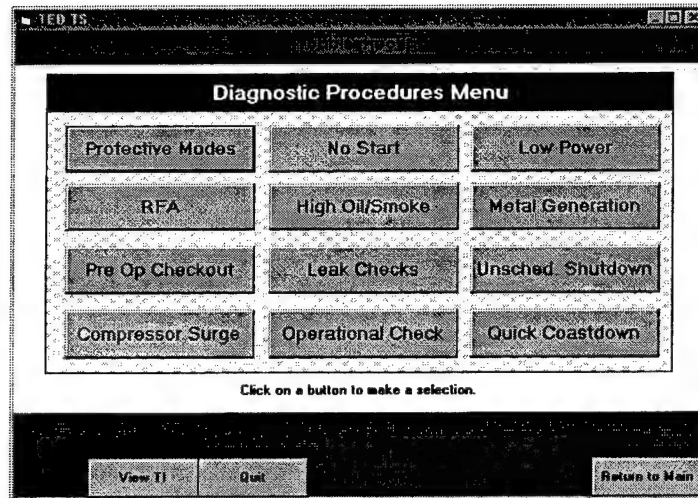


Figure 4. Sample TED Menu.

The rapid expansion of WEB services also provides an opportunity to create a dynamic diagnostic tool such as TED. A WEB page was designed by the TED team (<http://rpstl.arl.mil/ted.html>) to act as a reference source and help desk. Current TED information can be found at this WEB site. Every day, new techniques are developed, which allow the WEB participant to accomplish so much more through this dynamic environment.

As WEB services improve, on-line diagnostics will become possible. This will provide the mechanic with up-to-date diagnostic capabilities and provide a greater dynamic database to extract/record information that is currently spread by word of mouth or through manual updates. A WEB diagnostic tool for TED is already being researched, and a prototype will likely be developed within the next few years. This will, in turn, pay off through more efficient diagnostics and smarter mechanics from all world locations in contact with each other through the WEB.

6. References

1. De Mukler, R. V., C. Van Noortijkm, and H. O. Kerkmeester. "Knowledge Systems and Law—The JURICAS Project." *Proceedings of the Third International Conference on Logic, Informatics, Law (Legal Expert Systems)*, 1989.
2. Mikkonen, E., and T. Leinonen. "Expert System for Safety Inspection in Logging." IUFRO Div. 3.04.01 meeting. In Miscellaneous Report no. 385. Maine Agr. and For. Exp. Station. University of Maine, 1994.
3. Rosenberg, J. M. *Dictionary of Artificial Intelligence and Robotics*. New York: Wiley, 1986.
4. Textron Lycoming Internal Memo, 1988.
5. Textron Lycoming Internal Memo, 1989.
6. Georgeff, M. P., and A. L. Lansky. "Procedural Expert Systems." *Proceedings of the 8th International Joint Conference on AI*, 1983.
7. Georgeff, M. P., and A. L. Lansky. "A System of Reasoning in Dynamic Domains: Fault Diagnostics on the Space Shuttle." Technical Notes 375, SRI International, 1986.
8. Advanced Decision Systems. "Procedural Reasoning Systems." Working Paper, TR-ADS-0015, October 1988.
9. Klock, M., B. Rauch, S. Sperry, and A. Himes. "Visual Expert Guide." Visual Expert Manual, Version 2.21, 1994.
10. Weiss, S. M., and C. A. Kulikowski. *A Practical Guide to Designing Expert Systems*. Totowa, NJ: Rowman and Allanheld, 1984.
11. Boehm, B. W. "A Spiral Model of Software Development and Enhancement." ACM Software Engineering Notes, 1986.

INTENTIONALLY LEFT BLANK.

NO. OF
COPIES ORGANIZATION

2 DEFENSE TECHNICAL
INFORMATION CENTER
DTIC DDA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 HQDA
DAMO FDQ
DENNIS SCHMIDT
400 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 CECOM
SP & TRRSTRL COMMCTN DIV
AMSEL RD ST MC M
H SOICHER
FT MONMOUTH NJ 07703-5203

1 PRIN DPTY FOR TCHNLGY HQ
US ARMY MATCOM
AMCDCG T
M FISETTE
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 PRIN DPTY FOR ACQUSTN HQS
US ARMY MATCOM
AMCDCG A
D ADAMS
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 DPTY CG FOR RDE HQS
US ARMY MATCOM
AMCRD
BG BEAUCHAMP
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 DPTY ASSIST SCY FOR R&T
SARD TT T KILLION
THE PENTAGON
WASHINGTON DC 20310-0103

1 OSD
OUSD(A&T)/ODDDR&E(R)
J LUPO
THE PENTAGON
WASHINGTON DC 20301-7100

NO. OF
COPIES ORGANIZATION

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS AT AUSTIN
PO BOX 202797
AUSTIN TX 78720-2797

1 USAASA
MOAS AI W PARRON
9325 GUNSTON RD STE N319
FT BELVOIR VA 22060-5582

1 CECOM
PM GPS COL S YOUNG
FT MONMOUTH NJ 07703

1 GPS JOINT PROG OFC DIR
COL J CLAY
2435 VELA WAY STE 1613
LOS ANGELES AFB CA 90245-5500

1 ELECTRONIC SYS DIV DIR
CECOM RDEC
J NIEMELA
FT MONMOUTH NJ 07703

3 DARPA
L STOTTS
J PENNELLA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 USAF SMC/CED
DMA/JPO
M ISON
2435 VELA WAY STE 1613
LOS ANGELES AFB CA
90245-5500

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MDN A MAJ DON ENGEN
THAYER HALL
WEST POINT NY 10996-1786

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TP
2800 POWDER MILL RD
ADELPHI MD 20783-1145

NO. OF
COPIES ORGANIZATION

- 1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TA
2800 POWDER MILL RD
ADELPHI MD 20783-1145
- 3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

- 4 DIR USARL
AMSRL CI LP (305)

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1998	3. REPORT TYPE AND DATES COVERED Final, Jan - Aug 97	
4. TITLE AND SUBTITLE Turbine Engine Diagnostics (TED): A Practical Application of a Diagnostic Expert System			5. FUNDING NUMBERS 622783.Y10	
6. AUTHOR(S) Holly Ingham, Richard Helfman, Timothy Hanratty, John Dumer, and Ed Baur				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-IS-CI Aberdeen Proving Ground, MD 21005-5066			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-SR-60	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>Turbine engine diagnostics (TED) is a diagnostic expert system that aids the M1 Abrams' mechanic in finding and fixing problems in the AGT-1500 turbine engine. TED was designed to provide the apprentice mechanic the ability to diagnose and repair the turbine engine like an expert mechanic. This report discusses the reasoning method used in TED, called the procedural reasoning system (PRS), as well as various design considerations throughout the life of the project. The expert system was designed and built by the U.S. Army Research Laboratory (ARL) and the U.S. Army Ordnance Center and School (USAOC&S). TED has been fielded to both the Active Army and the National Guard.</p>				
14. SUBJECT TERMS artificial intelligence, expert system, procedural reasoning, diagnostics, turbine engine			15. NUMBER OF PAGES 20	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-SR-60 (Ingham) Date of Report January 1998

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT
ADDRESS

Organization

Name

E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD
ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
US ARMY RESEARCH LABORATORY
ATTN AMSRL IS CI
ABERDEEN PROVING GROUND MD 21005-5067



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

